

## Using Firefly Tools to Enhance Archive Web Pages

William Roby,<sup>1</sup> Xiuqin Wu,<sup>1</sup> Loi Ly,<sup>1</sup> and Tatiana Goldina<sup>2</sup>

<sup>1</sup>*Caltech, Mail Code 100-22, Pasadena, CA, 91125-2200, USA*

<sup>2</sup>*3038 Old Glenview Rd., Wilmette, IL 60091, USA*

**Abstract.** Astronomy web developers are looking for fast and powerful HTML 5/AJAX tools to enhance their web archives. We are exploring ways to make this easier for the developer. How could you have a full FITS visualizer or a Web 2.0 table that supports paging, sorting, and filtering in your web page in 10 minutes? Can it be done without even installing any software or maintaining a server? Firefly is a powerful, configurable system for building web-based user interfaces to access astronomy science archives. It has been in production for the past three years. Recently, we have made some of the advanced components available through very simple JavaScript calls. This allows a web developer, without any significant knowledge of Firefly, to have FITS visualizers, advanced table display, and spectrum plots on their web pages with minimal learning curve. Because we use cross-site JSONP, installing a server is not necessary. Web sites that use these tools can be created in minutes.

Firefly was created in IRSA, the NASA/IPAC Infrared Science Archive (<http://irsa.ipac.caltech.edu>). We are using Firefly to serve many projects including Spitzer, Planck, WISE, PTF, LSST and others.

### 1. Firefly Overview

Firefly is IRSA's new web-based system to access science archives. It provides new ground-breaking data visualization capabilities not available in many web-based archives. Our efforts to create a tightly integrated and user-friendly interface has generated numerous positive feedback from scientists. Users can browse and understand large volumes of archive data in a fraction of the time it took in the past.

AJAX technology gives us an opportunity to radically add value to basic data search and retrieval. In addition to searching and downloading, scientists can study, visualize, investigate, and begin to do science while browsing the archive. Not only can these advanced capabilities be an incredible time saver, but also improve the quality of the research.

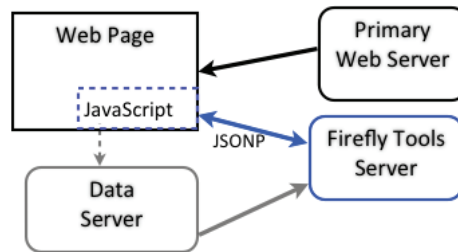
Firefly is being reused for multiple archives because we make all the components configurable using XML. This has allowed us to now implement additional archive systems at a fraction of the cost of the first system, the Spitzer Heritage Archive.

Firefly is implemented as a web application. This differs from a web page. With a web page, every time the display changes a new page is loaded. With a web application, there is just one page that is completely controlled by the JavaScript much like a desktop application. This approach does have some limitations. If you wish to take advantage of Firefly you must work completely in the Firefly environment. At IRSA, we want to provide cutting-edge web applications and enhance our existing web pages. Since we cannot retrofit the Firefly into existing web pages, we wrote the Firefly Tools interface to provide a solution to these limitations.

Firefly Tools exposes the most powerful components of Firefly in a way that can be used by any web page with no prerequisites. It allows any web developer access to Firefly's FITS visualizers or Table Tool with just a very few lines of JavaScript. The goal is to make these tools very easy to use with only a 10 minute learning curve.

An important feature is that the Firefly Tools server can be installed cross-site. In other words, it is not required to be on the same server as the web page. Firefly Tools can do this because it uses JSONP for the server communication. This allows Firefly Tools not to be limited by the server's Same Origin Policy and to give the developer a lot of flexibility. The web developer does not have to do any installation, but can simply just start using Firefly Tools. (see Figure 1)

Figure 1.: Firefly Tools server works in the web page by using cross-site scripting and communicating using JSONP



### 1.1. Major Firefly Tools Components

*Fits Visualization.* Firefly provides a first-class FITS visualization on the Web without any plugins. All of the significant components you would expect to see in a basic desktop FITS application are available with any FITS file that Firefly displays.

*Tabular Display.* Firefly has implemented Excel-like tables on the webpage. In an easy, interactive way, a user can sort the results, filter the data on multiple constraints, hide or show columns or select data for download or visualization. The Firefly Tools server is optimized to show very large tables without significant performance degradation. In addition, Firefly Tools also allows the user to easily connect tables to plots to create more integrated and powerful behavior.

*2D Line Graphs.* Firefly shows 2D line graphs interactively so that a user can read the data point values as he moves his mouse around or zooms in to investigate the data at a finer level. These graphs are used for spectrum or plotting table columns.

### 1.2. Examples

```

<script type="text/javascript" language='javascript'
      src='http://irsa.ipac.caltech.edu/fftools/fftools.nocache.js'> </script>
  
```

To load firefly Tools, you should include the above declaration in your HTML file. When Firefly Tools completes its loading, it calls a JavaScript function `onFireflyLoaded()`. Firefly Tools works by placing its components in a HTML `<div>` element with a specified ID. The following are several examples of what can be done in the `onFireflyLoaded()` function. All test data can be found at <http://web.ipac.caltech.edu/staff/roby/demo>.

*Note-* For brevity some of the HTML boiler plate code have been removed in the examples.

*Example 1 - Plotting One FITS Image.*

```
<div id="plot" style="width: 350px; height: 350px;"></div>

function onFireflyLoaded() {
  var iv2= firefly.makeImageViewer("plot");
  iv2.plot({
    "Title"      : "Example FITS Image",
    "ColorTable" : "16",
    "RangeValues": firefly.serializeRangeValues("Sigma", -2, 8, "Linear"),
    "URL"        : "http://web.ipac.caltech.edu/staff/roby/demo/wise-m31-level1-3.fits");
}
```

See result on the left hand side of Figure 2).

*Example 2 and 3 - Plotting Table and Image.* It is very easy to plot a table. The table can be used by itself or you can attach an image viewer to the table. The source for the table tool can be an IPAC table, a CSV, or a TSV file.

The following examples creates a side-by-side HTML layout. The table is plotted in the <div> labled "table2Here" and the related image is plotting in the <div> labled "previewHere". The table contains a column with the URL of an image related for that row. Every time the user clicks on a row the image will update. See the result on the left hand side of Figure 3.

```
<div style="white-space: nowrap;">
  <div id="table2Here" style="display:inline-block; width: 600px; height: 250px;
    margin : 5px 8px 0px 10px; border: solid 1px;"></div>
  <div id="previewHere" style="display:inline-block;
    width: 250px; height: 250px; border: solid 1px;"></div>
</div>

function onFireflyLoaded() {
  var table2Data={"source":"http://web.ipac.caltech.edu/staff/roby/demo/test-table4.tbl"};
  firefly.showTable(table2Data, "table2Here");
  firefly.addDataViewer( {"DataSource" : "URL",
    "DataColumn" : "FITS",
    "MinSize"    : "100x100",
    "ColorTable" : "1",
    "QUERY_ID"   : "table2Here" }, "previewHere" );
}
```

Using a similar approach you can create a coverage plot from a table that has columns with the four corners of each image. The HTML <div> declaration is omitted for brevity. See result on the right hand side of Figure 3.

```
function onFireflyLoaded() {
  var table1Data={"source":"http://web.ipac.caltech.edu/staff/roby/demo/WiseDemoTable.tbl"};
  firefly.showTable(table1Data, "tableHere");
  firefly.addCoveragePlot({"QUERY_ID" : "tableHere",
    "MinSize" : "100x100" }, "coverageHere" );
}
```

*Example 4 - Plotting related images.* In this example, we create four image viewers. Each belong to the same group ("wise-group"). We then set some global parameters so all the plots display the same. Now we plot each of the four images by specifying the URL of the FITS file. By doing this, all the plotting controls will work on all four images simultaneously. The HTML <div> declaration is omitted for brevity. See the result on right hand side of Figure 2.

```
function onFireflyLoaded() {
  var w1= firefly.makeImageViewer("w1", "wise-group");
  var w2= firefly.makeImageViewer("w2", "wise-group");
  var w3= firefly.makeImageViewer("w3", "wise-group");
```

```

var w4= firefly.makeImageViewer("w4", "wise-group");

firefly.setGlobalDefaultParams({ "ZoomType"      : "TO_WIDTH",
                                "ColorTable"     : "8",
                                "ZoomToWidth"    : "250" });

w1.plot({ "URL"   : "http://web.ipac.caltech.edu/staff/roby/demo/wise-m51-band1.fits",
          "Title": "Wise band 1" });
w2.plot({ "URL"   : "http://web.ipac.caltech.edu/staff/roby/demo/wise-m51-band2.fits",
          "Title": "Wise band 2" });
w3.plot({ "URL"   : "http://web.ipac.caltech.edu/staff/roby/demo/wise-m51-band3.fits",
          "Title": "Wise band 3" });
w4.plot({ "URL"   : "http://web.ipac.caltech.edu/staff/roby/demo/wise-m51-band4.fits",
          "Title": "Wise band 4" });
}

```

Figure 2.: *Left* - image is from Example 1, *Right* - group of images from Example 4

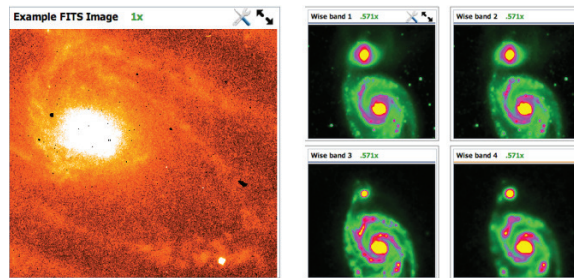
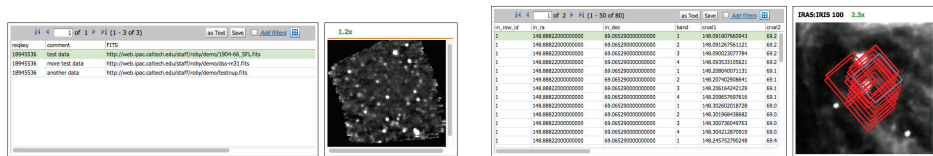


Figure 3.: *Left* - Example 2: Table and Image- Click on the table and the image updates. *Right* - Example 3: Table and Coverage- Click on the image or the coverage and the other updates.



### 1.2.1. More Information

If you would like more information, more examples or would like to try Firefly Tools yourself, please contact Trey Roby at [roby@caltech.edu](mailto:roby@caltech.edu) or 626-395-8681.